

Programmable Quantum Networks with Pure States

Alexander Yu. Vlasov

February 1, 2008

Abstract

Modern classical computing devices, except of simplest calculators, have von Neumann architecture, *i.e.*, a part of the memory is used for the program and a part for the data. It is likely, that analogues of such architecture are also desirable for the future applications in quantum computing, communications and control. It is also interesting for the modern theoretical research in the quantum information science and raises challenging questions about an experimental assessment of such a programmable models. Together with some progress in the given direction, such ideas encounter specific problems arising from the very essence of quantum laws. Currently are known two different ways to overcome such problems, sometime denoted as a stochastic and deterministic approach. The presented paper is devoted to the second one, that is also may be called *the programmable quantum networks with pure states*.

In the paper are discussed basic principles and theoretical models that can be used for the design of such nano-devices, *e.g.*, the conditional quantum dynamics, the Nielsen-Chuang “no-programming theorem,” the idea of deterministic and stochastic quantum gates arrays. Both programmable quantum networks with finite registers and hybrid models with continuous quantum variables are considered. As a basic model for the universal programmable quantum network with pure states and finite program register is chosen a “CONTROL-SHIFT” quantum processor architecture with three buses introduced in earlier works. It is shown also, that quantum cellular automata approach to the construction of an universal programmable quantum computer often may be considered as the particular case of such design.

1 Introduction

There are two almost independent ways of the classification of the quantum computational networks. In this paper and in many other works about the theory of quantum

computations are used models with quantum networks acting on pure states represented as vectors in Hilbert spaces [1]. It is also possible to consider mixed states and to use density matrices [2]. Such methods are especially useful in theory of open quantum systems, but for networks described in this paper it is enough to consider only pure states.

On the other hand, the *structure* of a quantum network may be specified using few different levels, which also have some analogue with the classical case. The simplest case — is a fixed network for the resolution of a particular task, *e.g.*, the quantum network for Shor’s factoring algorithm [3]. In the quantum case there are three different steps: the *preparation* of initial state, the quantum *evolution* described by given quantum network, and the *measurement* (Fig. 1a).

More difficult level — is a network with tuneable and interchangeable elements for managing with different tasks (Fig. 1b). It is common also for most experiments. Here is also present the initialization and the read-out (measurement), but between them may be considered different sequences of operations. Already in the earliest works [4] was raised a question about an universal set of such elementary operations, *the quantum gates*.

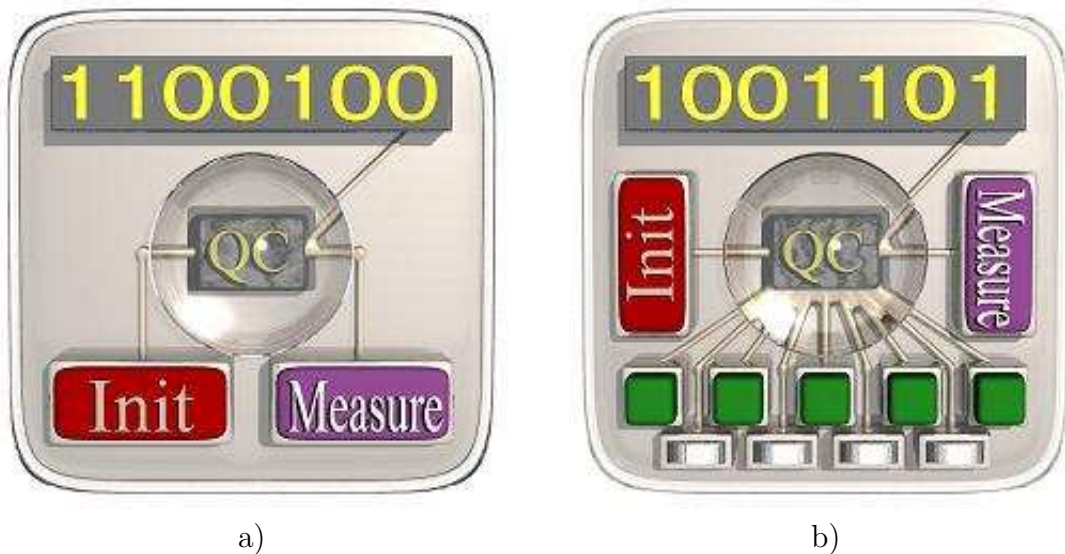


Figure 1: Usual schemes of quantum networks. a) The simplest quantum network with a fixed structure. b) The network with a set of gates for external control.

Even the second kind of network still rather resembles a quantum “calculator” Fig. 1b than a computer, because it is controlled via external manipulations instead of doing some program. It is especially essential for the quantum case, because such a

control here is described as a classical process and so there is an additional difficulty for the unified description of the whole system.

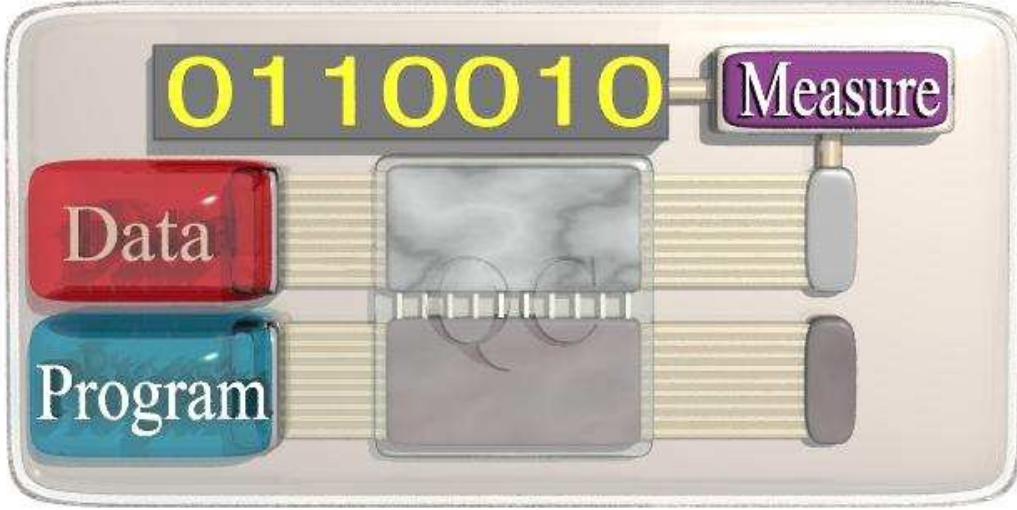


Figure 2: The programmable quantum network.

It is possible to consider a next level — the programmable quantum network with two subsystems: *the data* and *the program* Fig. 2. It is similar with von Neumann architecture of the classical computers [5].

With the usual notation of the quantum computation the programmable quantum networks U (with pure states) may be described as:

$$U(|D\rangle|II\rangle) = (\mathbf{u}_{II}|D\rangle)|II\rangle. \quad (1)$$

Here $|D\rangle$ and $|II\rangle$ are states of *data* and *program* registers before operation. After the application of the *fixed* unitary operator U , the state of the data register may be described as $\mathbf{u}_{II}|D\rangle$, *i.e.*, some operator \mathbf{u}_{II} is applied to the data state and it depends on the state $|II\rangle$ of the program register. State of the program register after the operation U is not changed (a more general case is represented in Eq. (1') below on the page 5).

It was found [6] that for the pure states Eq. (1) may be valid only if all different states $|II\rangle$ of the program register *are orthogonal*. *E.g.*, if there is some program (state) $|II_1\rangle$ for the implementation of an operator \mathbf{u}_1 and we need to implement another operator \mathbf{u}_2 using some program $|II_2\rangle$, then:

$$\mathbf{u}_1 \neq \mathbf{u}_2 \implies \langle II_1 | II_2 \rangle = 0. \quad (2)$$

The Eq. (2) was derived in [6] from Eq. (1) and unitarity of U (see Sec. 2.2).

It was considered in [6] as some kind of *the no-go theorem* for universality of a programmable quantum network, because in such a case dimension of the Hilbert space for the program register must be equal to the number of different programs, *i.e.*, unitary operators, but for the *exactly universal* quantum computer there are *infinite* number of such programs.

There are a few important no-go results in the theory of quantum computing and most known — is a *no-cloning* theorem [7]. The no-cloning theorem may be deduced from *linearity* of quantum mechanics, and the just mentioned “no-programming” problem even more subtle, *e.g.*, exists a *linear non-unitary* operator satisfying Eq. (1) with dimension of the program register only in two times bigger than the data one [8] (see Sec. 2.3). The problem with infinite dimension of the program register in the universal programmable quantum network appears due to the orthogonality of the different program states Eq. (2) derived from unitarity of U .

The no-programming problem initially had some constructive implication, because in [6] was suggested a special new kind of *stochastic* quantum networks to resolve the problem of universality. In the *stochastic programmable quantum gate arrays* [6] the size of a program is also only in two times bigger than the data, but the result of calculation is *non-deterministic*. The stochastic, probabilistic design described in many works [6, 9, 10, 11, 13] and will not be discussed here with details.

The models with measurements and probabilities make actual using programmable networks with mixed states and density matrices [12, 14], but it also deserves a separate account and not presented here.

In the present paper is considered the deterministic design with pure states and it is shown, that the problem with universality has rather formal meaning. From the one hand, only *finite* number of different operators are necessary for *the universality in the approximate sense* often used by default already in the earliest works about the quantum networks [1, 4, 15] and so it is possible to construct an universal (in the approximate sense) programmable quantum network with finite program register [16, 17, 18, 19].

From the other hand, even if to use the notion of *the exact universality* [20] with the infinite number of programs, it is possible to use *the quantum computations with continuous variables* [21] for generalization of considered models for the infinite-dimensional program register [18].

The models of programmable quantum networks with pure states considered in this paper have the direct analogue with *the conditional quantum dynamics* [22].

Contents

In Sec. 2 is represented *the formal theory of programmable quantum networks with pure states*. Definitions are revisited in Sec. 2.1. Limitations due to laws of quantum mechanics and the conception of the deterministic and stochastic programmable quantum gate arrays are discussed in Sec. 2.2. Quantum networks presented in this paper are *deterministic* in such classification, but some specific points relevant to the stochastic design may be found in Sec. 2.3. The deterministic design is originated from the idea of conditional quantum dynamics recalled in Sec. 2.4. In present paper are also mainly used quantum systems with finite-dimensional Hilbert spaces, but a hybrid model with continuous quantum variables is also not omitted and may be found in Sec. 2.5. The design of a programmable **Control-Shift** network is represented finally in Sec. 2.6 and is used as a basic model for the rest of the paper.

In the Sec. 3 are discussed *more concrete models*. The theory of universality in the quantum computations and control is recollected in Sec. 3.1. All necessary Hamiltonians for the programmable **Control-Shift** network with the universal set of quantum gates are constructed in Sec. 3.2. The relation with the theory of quantum cellular automata is demonstrated in Sec. 3.3.

2 Formal theory of programmable quantum networks

2.1 Definition

Let us return to the main equation describing a programmable quantum network Eq. (1). It is also may be written in more general form, when the state of the program register may be changed after the operation:

$$U: (|D\rangle \otimes |II\rangle) \mapsto (\mathbf{u}_II|D\rangle) \otimes |II'\rangle. \quad (1')$$

The scheme of a network for such transformation is depicted on Fig. 3. In the Eq. (1') is used expanded notation with the tensor product $|D\rangle \otimes |II\rangle$ often omitted for simplicity in expressions like $|D\rangle|II\rangle$ or $|D, II\rangle$.

Sometime it is convenient to exchange program and data registers and to write

$$U: (|II\rangle \otimes |D\rangle) \mapsto |II'\rangle \otimes (\mathbf{u}_II|D\rangle). \quad (1'')$$

2.2 Limitations due to quantum laws

The Eq. (1) and Eq. (1') was analyzed in [6], there such kind of networks was called *the programmable quantum gate arrays*. Sometimes they are also called *quantum processors*



Figure 3: Scheme of a programmable quantum network

[8, 11, 12, 13, 14, 17, 18, 19]. In Eq. (1') the state of the program register $|\Pi'\rangle$ after the operation should not depend on the state of the data register, because for two different data states it could be written

$$\begin{aligned} U(|D_1\rangle \otimes |\Pi\rangle) &= (\mathbf{u}_\Pi |D_1\rangle) \otimes |\Pi'_1\rangle \\ U(|D_2\rangle \otimes |\Pi\rangle) &= (\mathbf{u}_\Pi |D_2\rangle) \otimes |\Pi'_2\rangle. \end{aligned} \quad (3)$$

The unitary operator U does not change the scalar product and it is possible to write

$$\langle D_1 | D_2 \rangle \underbrace{\langle \Pi | \Pi \rangle}_1 = \langle D_1 | D_2 \rangle \langle \Pi'_1 | \Pi'_2 \rangle. \quad (4)$$

If states $|D_1\rangle$ and $|D_2\rangle$ are not orthogonal $\langle D_1 | D_2 \rangle \neq 0$, Eq. (4) is satisfied only for $\langle \Pi'_1 | \Pi'_2 \rangle = 1$ and so $|\Pi'_1\rangle = |\Pi'_2\rangle$.

For $\langle D_1 | D_2 \rangle = 0$ such an argument does not work, because Eq. (4) is always true. It is really possible to resolve Eq. (3) for orthogonal data states, but in such a case for different states $|\Pi'_1\rangle$ and $|\Pi'_2\rangle$ due to linearity it may be written:

$$U((\alpha|D_1\rangle + \beta|D_2\rangle) \otimes |\Pi\rangle) = \alpha(\mathbf{u}_\Pi |D_1\rangle) \otimes |\Pi'_1\rangle + \beta(\mathbf{u}_\Pi |D_2\rangle) \otimes |\Pi'_2\rangle,$$

so the state of program and data registers may be entangled and the equation does not have required form. Only for $|\Pi'_1\rangle = |\Pi'_2\rangle = |\Pi'\rangle$ the equation may be reduced to the proper form Eq. (1') for the arbitrary superposition of data states

$$U((\alpha|D_1\rangle + \beta|D_2\rangle) \otimes |\Pi\rangle) = (\mathbf{u}_\Pi(\alpha|D_1\rangle + \beta|D_2\rangle)) \otimes |\Pi'\rangle.$$

In the [6] also was found another important consequence of the given structure of Eq. (1') — *the orthogonality of different program states*. Really, let as consider two

different programs $|II\rangle$ and $|\Xi\rangle$. It is possible to write

$$\begin{aligned} U(|D\rangle \otimes |II\rangle) &= (\mathbf{u}_{II}|D\rangle) \otimes |II'\rangle \\ U(|D\rangle \otimes |\Xi\rangle) &= (\mathbf{u}_{\Xi}|D\rangle) \otimes |\Xi'\rangle. \end{aligned} \quad (5)$$

Due to unitarity of U

$$\langle II|\Xi\rangle = \langle D|\mathbf{u}_{II}\mathbf{u}_{\Xi}|D\rangle \langle II'|\Xi'\rangle \quad (6)$$

In [6] was noted, that because only the term $\langle D|\mathbf{u}_{II}\mathbf{u}_{\Xi}|D\rangle$ in Eq. (6) depends on the state $|D\rangle$, it may be resolved only for $\langle II|\Xi\rangle = 0 = \langle II'|\Xi'\rangle$ *i.e.*, if all *different programs correspond to orthogonal states*. So the orthogonality condition Eq. (2) mentioned in Sec. 1 is proved.

• *Note:* It should be mentioned, that the question about behavior of terms like $\langle D|\mathbf{u}|D\rangle$ maybe denotes additional discussions, but it is above a scope of presented paper. Say, it is possible to consider a real-valued analogue of the qubit [23, 24], a “*rebit*.” It is a vector $|R\rangle$ in the two-dimensional real vector space (plane). The operators \mathbf{u} now correspond to rotations of the plane. In such a case $\langle R|\mathbf{u}|R\rangle = \cos(\varphi)$ *does not depend* on the state of the rebit (here φ is the angle of rotation).

Due to the orthogonality property Eq. (2) for a program register in Eq. (1), the dimension of the Hilbert space is equivalent to the number of different operators \mathbf{u}_{II} . For the exact universality we formally must have possibility to apply the infinite number of different unitary operators \mathbf{u} . It was considered in [6] as some disadvantage and it was suggested idea of “probabilistic” (non-deterministic) programmable quantum gate arrays.

2.3 Non-unitary linear operators and non-deterministic networks

It should be mentioned, that the no-go result about the exactly universal deterministic programmable quantum network may be considered as a more subtle limitation in comparison with a famous quantum no-cloning theorem [7]. The no-cloning theorem may be derived from the linearity of the quantum mechanics, but the consideration above also uses the *unitarity*. It is essential, because exist linear, but non-unitary operators satisfying Eq. (1') with the finite-dimensional program register.

If a data register is described by some state vector $|D\rangle \in \mathcal{H}$ with components D_i , then the minimal program register must contain the matrix of coefficients for the operator A_{ij} and so in the simplest case such a register is two times bigger than the data one $|A\rangle \in \mathcal{H} \otimes \mathcal{H}$. Here due to the usual law of composition of quantum systems for N -dimensional data register such a program register is N^2 -dimensional.

Let $|A\rangle = \sum A_{ij}|i\rangle|j\rangle$ is the state of the program encoding the operator $\mathbf{A} = \sum A_{ij}|i\rangle\langle j|$, *viz* $|A\rangle = \sum_j |j\rangle \mathbf{A}|j\rangle$ and \mathbf{M} is the linear non-unitary operator defined on

the basis as

$$\mathbf{M}: |k\rangle|i\rangle|j\rangle \mapsto \delta_{jk}|i\rangle|0\rangle|0\rangle. \quad (7)$$

Then a specific kind of Eq. (1') is valid

$$\mathbf{M}(|\psi\rangle|A\rangle) = |\mathbf{A}\psi\rangle|0,0\rangle, \quad (8)$$

because $(\mathbf{A}\psi)_i = \sum_j A_{ij}\psi_j = \sum_{jk} A_{ij}\delta_{jk}\psi_k$. In fact, the \mathbf{M} is the operator of the matrix multiplication rewritten in a specific way.

The idea has some relation with the non-deterministic design [6]. Definition of the state $|U\rangle$ representing an unitary operator \mathbf{U} may be simply changed, to ensure the unit norm

$$|U\rangle = \frac{1}{\sqrt{N}} \sum_j |j\rangle(\mathbf{U}|j\rangle). \quad (9)$$

For the qubit (two-dimensional Hilbert space) it coincides with the definition in [6]. It is possible to rewrite $|\psi\rangle|U\rangle$ using the Bell basis $\Phi_{\pm} = \frac{\sqrt{2}}{2}(|00\rangle \pm |11\rangle)$, $\Psi_{\pm} = \frac{\sqrt{2}}{2}(|01\rangle \pm |10\rangle)$ for the first two qubits [6]

$$|\psi\rangle|U\rangle = \frac{1}{2}(|\Phi^+\rangle\mathbf{U}|\psi\rangle + |\Psi^+\rangle\mathbf{U}\sigma_x|\psi\rangle + i|\Psi^-\rangle\mathbf{U}\sigma_y|\psi\rangle + |\Phi^-\rangle\mathbf{U}\sigma_z|\psi\rangle). \quad (10)$$

With minimal modification such expression may be converted to the expansion of some *unitary* operator $\mathbf{G} = \mathbf{L} + \mathbf{R}$, there \mathbf{L} is the non-unitary linear “programming” operator and \mathbf{R} is the residue $\mathbf{R} = \mathbf{G} - \mathbf{L}$

$$\mathbf{G}(|\psi\rangle|U\rangle) = (\mathbf{U}|\psi\rangle)|0,0\rangle + \mathbf{R}(|\psi\rangle|U\rangle). \quad (11)$$

So the non-unitary linear operator at least formally presents in the model of the non-deterministic programmable quantum gate array, but further discussion on the question is outside of the scope of this paper.

2.4 Conditional quantum dynamics

Let us return to the programmable quantum networks denoted in [6] as *deterministic*. The result about the orthogonality of different program states also has some positive implications and simplifies description. It is possible to choose the states corresponding to different programs as a new basis and to write instead of $|\Pi_k\rangle$ simply $|k\rangle$. Structure of the operator \mathbf{U} Eq. (1) becomes quite clear in such notation, it is *the conditional quantum dynamics* [15] described even earlier, than the programmable quantum gate arrays mentioned above.

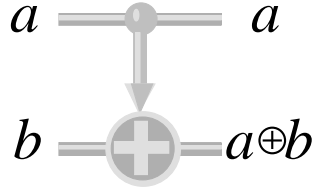
Such an unitary operator U for orthogonal $|k\rangle$ from m -dimensional Hilbert space may be written [15]

$$U = \sum_{k=0}^{m-1} |k\rangle\langle k| \otimes \mathbf{u}_k, \quad U(|k\rangle|D\rangle) = |k\rangle(\mathbf{u}_k|D\rangle). \quad (12)$$

In Eq. (12) program and data registers are exchanged in comparison with definition Eq. (1) or Eq. (1') and it corresponds to the alternative notation Eq. (1''). Such an order is used further in the paper for convenience (see *Note* on the page 11).

A simplest example — is the **controlled-NOT**, **c-NOT** gate.

It was used already in earliest papers about quantum computers [25] and also called *the measurement gate* [1]. It is the quantum version of a classical reversible gate with two input and two output states. The **c-NOT** applies **NOT** to a second state if and only if the first state is 1 (**TRUE**), *i.e.*, for Boolean variables a, b may be described as $(a, b) \mapsto (a, b \oplus a)$, where “ \oplus ” is addition modulo 2, $0 \oplus 0 = 1 \oplus 1 = 0$, $0 \oplus 1 = 1 \oplus 0 = 1$.



The quantum mechanical version is straightforward

$$\text{c-NOT}_{12}: |a\rangle|b\rangle \mapsto |a\rangle|b \oplus a\rangle. \quad (13)$$

Two possible instances of the **c-NOT** gate are depicted on Fig. 4. They also may be considered as representations of two different orders of program and data registers used in Eq. (12) and Eq. (1).

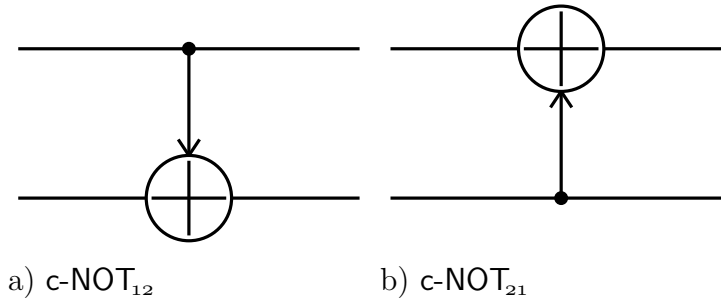


Figure 4: Schemes of **controlled-NOT** quantum gates. a) The first qubit — is control $|a\rangle|b\rangle \mapsto |a\rangle|b \oplus a\rangle$. b) The second qubit — is control $|a\rangle|b\rangle \mapsto |b \oplus a\rangle|b\rangle$.

It is useful to consider the simple case to show principle of matrix representation

of such operators. The action of the **c-NOT** gate for the basis may be written as

$$\begin{aligned}
|0\rangle|0\rangle &\mapsto |0\rangle|0\rangle \\
|0\rangle|1\rangle &\mapsto |0\rangle|1\rangle \\
|1\rangle|0\rangle &\mapsto |1\rangle|1\rangle \\
|1\rangle|1\rangle &\mapsto |1\rangle|0\rangle
\end{aligned} \tag{14}$$

so the **c-NOT** gate simply exchanges two last vectors of the basis $|10\rangle \leftrightarrow |11\rangle$ and the matrix of the operation may be written as

$$\mathbf{c-NOT}_{12} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \tag{15}$$

Unlike the classical case, the difference between **c-NOT**₁₂ and **c-NOT**₂₁ quantum gates Fig. 4 is rather formal — it is enough to change bases for the both systems to convert one gate to another and so both qubits formally are equal in such operation [27, 28]. The new basis is

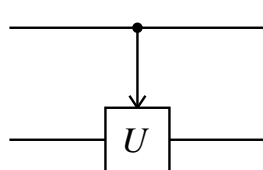
$$|+\rangle = \frac{\sqrt{2}}{2} (|0\rangle + |1\rangle), \quad |-\rangle = \frac{\sqrt{2}}{2} (|0\rangle - |1\rangle). \tag{16}$$

In the new $|\pm\rangle$ basis **c-NOT**₁₂ Eq. (14) is rewritten

$$\begin{aligned}
|+\rangle|+\rangle &= \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) \mapsto \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) = |+\rangle|+\rangle \\
|+\rangle|-\rangle &= \frac{1}{2} (|00\rangle - |01\rangle + |10\rangle - |11\rangle) \mapsto \frac{1}{2} (|00\rangle - |01\rangle - |10\rangle + |11\rangle) = |-\rangle|-\rangle \\
|-\rangle|+\rangle &= \frac{1}{2} (|00\rangle + |01\rangle - |10\rangle - |11\rangle) \mapsto \frac{1}{2} (|00\rangle + |01\rangle - |10\rangle - |11\rangle) = |-\rangle|+\rangle \\
|-\rangle|-\rangle &= \frac{1}{2} (|00\rangle - |01\rangle - |10\rangle + |11\rangle) \mapsto \frac{1}{2} (|00\rangle - |01\rangle + |10\rangle - |11\rangle) = |+\rangle|-\rangle
\end{aligned} \tag{17}$$

The Eq. (17) show, that a first element is swapped if and only if the second one is $|-\rangle$, it is just **c-NOT**₂₁ written in the new bases $|\pm\rangle|\pm\rangle$. So there is no clear distinction between the control and the controlled system. It is shown further, that such situation may be not valid for more difficult cases.

Similarly with **controlled-NOT** it is possible for an arbitrary gate $\mathbf{U} = \begin{pmatrix} U_{00} & U_{01} \\ U_{10} & U_{11} \end{pmatrix}$ to define the **controlled-U** gate sometimes denoted as $\Lambda_1(\mathbf{U})$ [26]



$$\Lambda_1(\mathbf{U}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{00} & U_{01} \\ 0 & 0 & U_{10} & U_{11} \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \tag{18}$$

Using the same principle, the sum used in Eq. (12) may be written as the block-diagonal matrix

$$U = \begin{pmatrix} \mathbf{u}_0 & & & 0 \\ & \mathbf{u}_1 & & \\ & & \ddots & \\ 0 & & & \mathbf{u}_{m-1} \end{pmatrix}. \quad (19)$$

• *Note:* For the m -dimensional program register and the n -dimensional data register such U is $mn \times mn$ matrix with only nonzero elements are $n \times n$ blocks with matrices \mathbf{u}_k on diagonal of U . The understanding notation Eq. (19) may be used if the program register is the first system like in Eq. (1'') and Eq. (12).

Due to the orthogonality condition Eq. (2) the dimension of the program register often much bigger $m \gg n$ or maybe even infinite and so, unlike the example with **c-NOT** gate, two systems used for data and program registers in general are not equivalent.

2.5 Infinite-dimensional program register

It was already mentioned, that for the universality in the exact sense the dimension of a program register should be infinite [6]. Let us consider such a case. Such a programmable network uses both continuous and discrete quantum variables for the program and the data registers respectively [18] and so may be considered as an example of the *hybrid* quantum network [29].

The system is *hybrid* also in other meaning [30], *i.e.*, it is the possible approach to the unified description of a quantum system controlled by some analogue parameters. It make some bridge with usual (non-programmable) quantum networks based on pseudo-classical description of “tuneable” gates depicted schematically on Fig. 1b.

For transition to the continuous variables it is possible to change the sum in Eq. (12) to the integral and write formally [18]

$$U = \int (|q\rangle\langle q| \otimes \mathbf{u}(q)) dq, \quad U(|q\rangle|D\rangle) = |q\rangle(\mathbf{u}(q)|D\rangle). \quad (20)$$

The Eq. (20) describes the family of gates (matrices) $\mathbf{u}(q)$ parameterized by some continuous variable q . Elements of the basis in infinite-dimensional Hilbert space are denoted as $|q\rangle$.

A standard example — is the space of functions $\psi(x)$ on a line and Dirac delta functions as the basis

$$|q\rangle = \delta(x - q), \quad \langle q|\psi(x)\rangle = \psi(q). \quad (21)$$

The sign \otimes of tensor product may be skipped in the expression like Eq. (20), because for such a hybrid case with continuous and discrete quantum variables there is simple

interpretation. The tensor product of the space of functions on a line and some finite-dimensional vector space may be represented as a space of multi-component functions on the line with values in the vector space. Say the both program and data registers may be encoded in a wave vector $|\Psi(x)\rangle$ of one particle.

The space of linear operators in such a case may be represented as a space of matrix-valued operators on the line. For example Eq. (20) may be expanded as

$$U(\psi(x)|D\rangle) = \int \delta(x-q)\psi(q)\mathbf{u}(q)|D\rangle dq = \psi(x)\mathbf{u}(x)|D\rangle. \quad (22)$$

Using the example with one particle, Eq. (22) may be considering as a process with “twisting along x ” of the wave function $|\Psi(x)\rangle$ depicted on Fig. 5.



Figure 5: A scheme of the process Eq. (22) with a “distributed” qubit.

It is also possible to use the momentum basis of the periodic functions

$$|\widetilde{p}\rangle = \exp(2\pi i p x). \quad (23)$$

With the new basis it is possible to write

$$\tilde{U} = \int (|\widetilde{p}\rangle\langle\widetilde{p}'|\mathbf{u}(p))dp, \quad \tilde{U}(|\widetilde{p}\rangle|D\rangle) = |\widetilde{p}'\rangle(\mathbf{u}(p)|D\rangle), \quad (24)$$

there is also taken into account the possibility of change of the momentum $p \rightarrow p'(p)$ (the program register) similarly with Eq. (1').

More details may be found in [18], let us only consider a visual example with “a scattering process” Fig. 6, there are used two different systems.

Before the scattering (Fig. 6, time t_1) the first system was described by a “monochrome” wave function $\exp(ikx)$ and the second one — is a localized qubit $|D\rangle$. After the “inelastic” scattering process (Fig. 6, time t_2) the first system may have a state with other $k' = k'(k)$ and the qubit changed a state to $\mathbf{u}(k)|D\rangle$.

Such a scattering process maybe too formal and abstract, but it provides the understanding analogue between the programmable quantum network and a more traditional design with a quantum system controlled by some external devices like lasers.

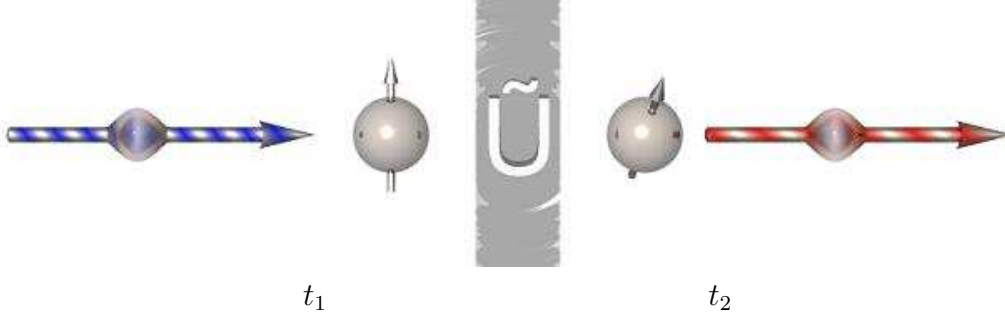


Figure 6: Illustration of Eq. (24) with scattering process.

It should be mentioned, that the considered model with continuous variables also provides some link with the stochastic programmable quantum gate arrays. It was already mentioned, that such a model produces a correct answer with some probability. For N -dimensional data register the probability is $1/N$, *i.e.*, 2^{-n} for a system with n qubits [6]. For some particular case of encoding $U(1)$ operations, it is possible to make the probability higher for the bigger program register with approach to the unit probability of success in the infinite limit [9, 10].

On the other hand, for the deterministic design it is also possible to consider the infinite-dimensional program register as a limit of finite-dimensional one. Sequences of finite-dimensional networks used for such limits have quite different properties for stochastic and deterministic approach. For the first one we may apply an arbitrary operation $U(1)$, similar with rotations on arbitrary angles $2\pi\phi$, but it succeeds only with some probability increasing with approach to the infinite limit.

For the second one, the operation is always successful, but we may apply only finite number of different operations $U(1)$, *i.e.*, rotations on some fixed angles $2\pi k/N$, and for the infinite limit the rotations cover the full circle. Despite of such difference for the finite case, *the infinite limits for both deterministic and stochastic networks for $U(1)$ operation are essentially the same*. A rather technical proof of the interesting fact may be found in [18] and is not reproduced here.

2.6 Programmable quantum networks with three buses

It should be mentioned, that Eq. (1) and all derivative equations considered here are still do not have the complete analogue with the idea of usual (von Neumann) computer architecture, because formally they are describe *only one step* of a program.

From the one hand, such a picture sometimes may be appropriate for the theory of quantum computing. On the other one, it is very limiting, because even in the standard definition of the universal set of quantum gates, it is used *the arbitrary composition* of

the gates from the set [1, 4, 15, 20, 26].

Such an idea of the universality may be implemented also for the programmable quantum networks with pure states [16, 17, 18, 19] discussed in this paper. The scheme Fig. 7 uses two operators and three buses.

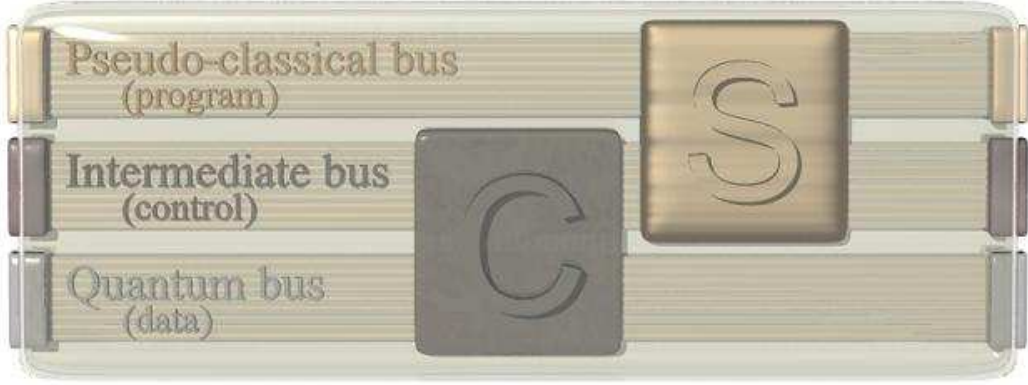


Figure 7: The programmable quantum network (processor) with three buses

The **C**, **Control** — is the already considered operator Eq. (1'') of *one computational step* with second (control) and third (quantum data) buses. The **S**, **Shift** operator must change the state $|k\rangle$ of a control bus after each step using first (program) and second buses.

The buses also may be called *pseudo-classical*, *intermediate* and *quantum*, because the quantum bus may contain arbitrary superposition of states, the intermediate bus should use only orthogonal states, but it is still may not be considered by an entirely classical way, because it is linked via **C** operator with the quantum bus and the specific character of such a design was already discussed above. But already for description of the pseudo-classical bus and the operator **S** it is at least formally enough to use the theory of reversible classical gates.

In principle, **S** may correspond to an arbitrary *reversible* classical program, but here is convenient to consider the simplest case with the *cyclic shift*. Let us suggest, that a data register is described by m -qubits, there is a finite set with $N = 2^n$ quantum gates, and for our purposes it is necessary to apply a sequence with L gates from the given set. For example we may consider the finite set of universal gates and the task of the approximation of an arbitrary gate with the necessary precision using up to L gates.

For the design with the cyclic shift operator **S** a program register must have size $(L - 1)n$ qubits and acts on Ln qubits of program and control registers as

$$S: |k_L, k_{L-1}, \dots, k_2\rangle |k_1\rangle \mapsto |k_1, k_L, \dots, k_3\rangle |k_2\rangle. \quad (25)$$

Action of C was already defined in Eq. (12)

$$C: |k\rangle|D\rangle \mapsto |k\rangle(\mathbf{u}_k|D\rangle). \quad (12')$$

and so the composition of C and S may be written

$$SC\left(|k_L, k_{L-1}, \dots, k_2\rangle|k_1\rangle|D\rangle\right) = |k_1, k_L, \dots, k_3\rangle|k_2\rangle(\mathbf{u}_k|D\rangle). \quad (26)$$

It is only one step. Let us denote the state of program and control registers as $|K\rangle = |k_L, k_{L-1}, \dots, k_2\rangle|k_1\rangle$. For L steps $|K\rangle$ returns to the initial state and so it is possible to write [16, 17, 18, 19]

$$(SC)^L: |K\rangle|D\rangle \mapsto |K\rangle(\mathbf{u}_L \cdots \mathbf{u}_2 \mathbf{u}_1|D\rangle) \quad (27)$$

Here the program bus should be rather compared with the *cyclic read-only memory* (ROM). A scheme of the programmable “Control-Shift” network with the cyclic shift register is depicted on Fig. 8.



Figure 8: A scheme of Control-Shift network

3 Models of programmable quantum networks

3.1 Universal sets of quantum gates

In the theory represented above were used rather abstract models. Standard relation between the scheme of a quantum gate and the evolution of the quantum system is depicted on Fig. 9. The input and output of the network — is really the same system at two different moments of time and the gate represents change of the state of the system during given period due to some interactions.

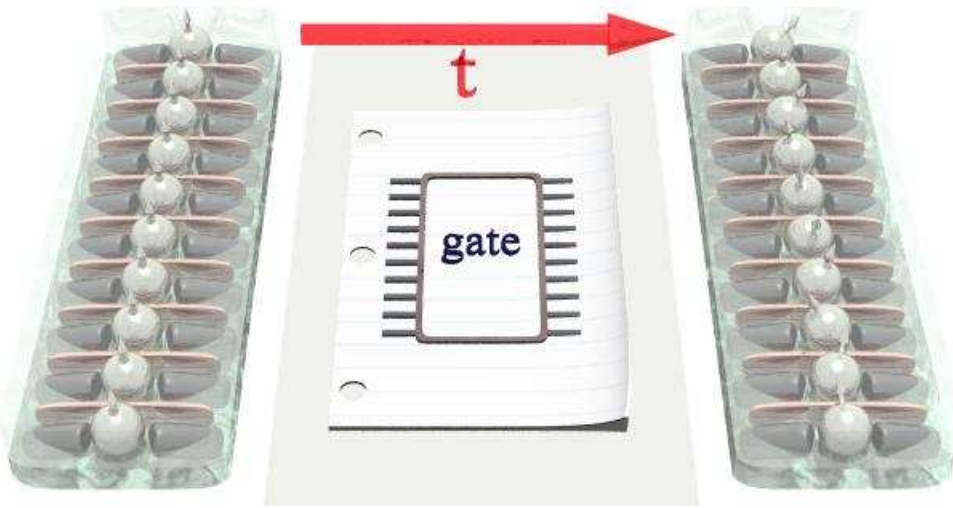


Figure 9: A scheme of a gate and evolution of a quantum system

Idea of decomposition using the universal set of quantum gates was already few times mentioned in the present paper, *e.g.*, it is relevant to the model of the **Control-Shift** network described by Eq. (27). Let us recall the basic principles of the theory of the universality in quantum computations and control.

There is so-called *infinitesimal* approach with the Lie algebras [15, 31, 32]. It is convenient also due to the direct relation with Hamiltonians of quantum systems. For the system with the constant Hamiltonian \mathbf{H} the evolution during the time t is described by the unitary operator (gate)

$$\mathbf{U} = \mathbf{U}(\tau) = \exp(-i \mathbf{H} \tau). \quad (28)$$

If two Hamiltonians \mathbf{H}_1 , \mathbf{H}_2 correspond to gates \mathbf{U}_1 , \mathbf{U}_2 , then

$$\mathbf{U}_1 \mathbf{U}_2 = \exp(-i \mathbf{H}_1 \tau) \exp(-i \mathbf{H}_2 \tau) = \exp(-i (\mathbf{H}_1 + \mathbf{H}_2) \tau) + O(\tau^2), \quad (29)$$

there $O(\tau^2)$ is an error of order τ^2 and so for small τ the sum of Hamiltonians corresponds to the composition of the gates. It is also quite standard to use an expression for commutators [15, 31, 32]

$$U_1 U_2 U_1^{-1} U_2^{-1} = \exp(-(\mathbf{H}_1 \mathbf{H}_2 - \mathbf{H}_2 \mathbf{H}_1) \tau^2) + O(\tau^3) \quad (30)$$

and so such a product with four terms is approximately equal to the action of a gate with a Hamiltonian $-i[\mathbf{H}_1, \mathbf{H}_2]$ and a parameter τ^2 , *i.e.*, if to consider $\tau = \sqrt{t}$ the Eq. (30) has precision $O(t^{1.5})$.

Due to Eq. (29) and Eq. (30), which valid for infinitesimal values $\tau \rightarrow 0$, it is possible to formulate a condition of universality using Hamiltonians [15, 31, 32]. The set of Hamiltonians \mathbf{H}_k corresponds to the universal set of gates, if it is possible to produce an arbitrary Hamiltonian using linear combinations of \mathbf{H}_k and they commutators of any order, *i.e.*, $i[\mathbf{H}_j, i[\mathbf{H}_k, \dots]]$.

Such definition is convenient for a test of universality, but for concrete tasks it may produce a problem due to equations for commutators. Already for Eq. (30) with τ^2 it is not clear, how to choose a parameter τ for the generation of a gate with a good precision, and the problem even worst for commutators of k -th order with τ^k .

There is a way to produce an expression with the first degree of τ instead of τ^2 in Eq. (30) and without error at all if to use special choice of Hamiltonians [33]. Let us consider a system with n -qubits. As a basis in the space of Hermitian matrices (Hamiltonians) may be used 4^n different tensor products of four Pauli matrices

$$\mathbf{H}_{\mathbf{j}} = \sigma_{j_1} \otimes \sigma_{j_2} \otimes \dots \otimes \sigma_{j_n}, \quad j_k = 0, \dots, 3 \quad (31)$$

where $\mathbf{j} = (j_1, j_2, \dots, j_n)$ is notation for multi-index and

$$\sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (32)$$

It is also widely used the alternative notation: $\mathbf{1}$, σ_x , σ_y , σ_z respectively.

All Hamiltonians Eq. (31) have properties

$$\mathbf{H}_{\mathbf{j}} \mathbf{H}_{\mathbf{k}} = \pm \mathbf{H}_{\mathbf{k}} \mathbf{H}_{\mathbf{j}}, \quad \mathbf{H}_{\mathbf{j}}^2 = \mathbf{1}. \quad (33)$$

Unity of the square in Eq. (33) ensures a simple expression for the exponent

$$\exp(i\phi \mathbf{H}_{\mathbf{k}}) = \cos(\phi) \mathbf{1} + i \sin(\phi) \mathbf{H}_{\mathbf{k}} \quad (34)$$

and using the identity

$$\exp(\mathbf{A}) \exp(\mathbf{B}) \exp(-\mathbf{A}) = \exp(\exp(\mathbf{A}) \mathbf{B} \exp(-\mathbf{A}))$$

it is possible to find the precise expression for the exponent of commutator

$$e^{-[\mathbf{H}_j, \mathbf{H}_k]\tau} = \begin{cases} 1, & \mathbf{H}_j \mathbf{H}_k = +\mathbf{H}_k \mathbf{H}_j \\ e^{i\frac{\pi}{4}\mathbf{H}_j} e^{2i\tau\mathbf{H}_k} e^{-i\frac{\pi}{4}\mathbf{H}_j}, & \mathbf{H}_j \mathbf{H}_k = -\mathbf{H}_k \mathbf{H}_j \end{cases}. \quad (35)$$

Unlike Eq. (30) the Eq. (35) contains the first degree of τ and for such a method of generation of arbitrary gates there are only errors related with Eq. (29), which have order $O(t^2)$.

It is well-known that one- and two-gates are enough for universality [15, 26, 31]. For the work with Hamiltonians like Eq. (31) with properties Eq. (33) it is convenient to use an universal set with $2n + 1$ elements [33]

$$\mathbf{Z}_0 = \sigma_3^{(1)} = \sigma_3 \otimes \underbrace{\sigma_0 \otimes \cdots \otimes \sigma_0}_{n-1}, \quad (36a)$$

$$\mathbf{Z}_1 = \sigma_3^{(2)} = \sigma_0 \otimes \sigma_3 \otimes \underbrace{\sigma_0 \otimes \cdots \otimes \sigma_0}_{n-2}, \quad (36b)$$

$$\mathbf{X}_k = \sigma_1^{(k+1)} = \underbrace{\sigma_0 \otimes \cdots \otimes \sigma_0}_k \otimes \sigma_1 \otimes \underbrace{\sigma_0 \otimes \cdots \otimes \sigma_0}_{n-k-1}, \quad (36c)$$

$$\mathbf{D}_k = \sigma_3^{(k+1)} \sigma_3^{(k+2)} = \underbrace{\sigma_0 \otimes \cdots \otimes \sigma_0}_k \otimes \sigma_3 \otimes \sigma_3 \otimes \underbrace{\sigma_0 \otimes \cdots \otimes \sigma_0}_{n-k-2}. \quad (36d)$$

Really the set of gates differs from suggested in [33] on nonessential change of the basis ($\sigma_1 \leftrightarrow \sigma_3$). After such exchange all two-qubit gates have diagonal form due to Eq. (36d). It should be reminded, that the gates are generated using Eq. (28) with given Hamiltonians Eq. (36). It may prevent some confusion, because all operators in form Eq. (31) are not only Hermitian, but also are unitary and so they are widely used as gates in many works.

There is also alternative approach to the universality, then instead of consideration of Hamiltonians and infinitesimal (small) transformations, it is considered a question, how to decompose precisely the given unitary matrix (with 4^n parameters for n qubits) on product of one and two-qubit gates. Standard choice is: all one-qubit gates together with **c**-NOT gate [34, 35, 36], *e.g.*, in [36] is suggested an algorithm for the decomposition with 4^n one-qubit gates and $4^n - 2^n$ **c**-NOT gates.

It should be mentioned, that **c**-NOT₁₂ gate also may be transformed to diagonal form, if to change the basis of the *second* qubit to $|\pm\rangle$ Eq. (16). Such transformation is described by the Hadamard matrix $\mathbf{H} = \frac{\sqrt{2}}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ and already was used above on page 10, there it was applied to *both* qubits for the transition between **c**-NOT₁₂ and

c-NOT₂₁ gates. After such diagonalization the Hamiltonian Eq. (15) may be written as

$$\text{d-NOT} = \bigwedge_1(\boldsymbol{\sigma}_3) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (37)$$

Really, any **controlled- \mathbf{U}** gate Eq. (18) may be transformed to diagonal form by some change of a basis, *i.e.*, using one-qubit operations with the second qubit. It is simply a basis there \mathbf{U} itself is diagonal.

3.2 Hamiltonians of elementary programmable gates

3.2.1 Control gates

Let us consider models of basic quantum gates, necessary for implementation of the universal programmable networks. In Sec. 3.1 it was shown, that it is enough to use “powers” of Pauli matrices

$$\exp(i\boldsymbol{\sigma}_k\tau) = \cos(\tau) + i\sin(\tau)\boldsymbol{\sigma}_k \quad (38)$$

together with some simple diagonal two-qubit gate. In the programmable quantum networks are used controlled gates and so basic elements are two- and three-gates.

The Hamiltonian of controlled versions of a gate may be found using the simple expression

$$e^{i\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{pmatrix}\tau} = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & e^{i\mathbf{H}\tau} \end{pmatrix}, \quad (39)$$

where $\mathbf{0}$, $\mathbf{1}$ are zero and identity matrices respectively. So Hamiltonians of two-gates for the controlled one-qubit operations used in the universal set Eq. (36) may be represented as

$$\mathbf{h}_X^c = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{h}_Z^c = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \quad (40)$$

The three-qubits Hamiltonian for the controlled diagonal gates Eq. (36d) is

$$\mathbf{h}_D^c = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (41)$$

Let us show, how to use instead of Eq. (41) only two-qubit Hamiltonians.

$$\begin{aligned} \mathbf{h}_D^c &= |1\rangle\langle 1| \otimes \sigma_z \otimes \sigma_z = \frac{1}{2}(\mathbf{1} - \sigma_z) \otimes \sigma_z \otimes \sigma_z \\ &= \frac{1}{2}(\sigma_z^{(2)}\sigma_z^{(3)} - \sigma_z^{(1)}\sigma_z^{(2)}\sigma_z^{(3)}) = \frac{1}{2}(\sigma_z^{(2)}\sigma_z^{(3)} - \frac{i}{2}[\sigma_x^{(1)}\sigma_z^{(2)}, \sigma_y^{(1)}\sigma_z^{(3)}]). \end{aligned} \quad (42)$$

If to consider the three-qubit gate generated by \mathbf{h}_D^c , due to the decomposition Eq. (42) and Eq. (35) it is possible to write

$$\begin{aligned} U_D^c(\tau) &= \exp(i\mathbf{h}_D^c\tau) = \exp(i\frac{\tau}{2}\sigma_z^{(2)}\sigma_z^{(3)}) \exp(-i\frac{\tau}{2}\sigma_z^{(1)}\sigma_z^{(2)}\sigma_z^{(3)}) \\ &= \exp(i\frac{\tau}{2}\sigma_z^{(2)}\sigma_z^{(3)}) \exp(i\frac{\pi}{4}\sigma_x^{(1)}\sigma_z^{(2)}) \exp(-i\frac{\tau}{2}\sigma_y^{(1)}\sigma_z^{(3)}) \exp(-i\frac{\pi}{4}\sigma_x^{(1)}\sigma_z^{(2)}). \end{aligned} \quad (43)$$

Let us consider also a Hamiltonian with two control qubits for implementation of all three Pauli matrices

$$\mathbf{h}_P^c = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & i & 0 & 0 \\ 0 & 0 & 0 & 0 & -i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}. \quad (44)$$

It is the simple example of a Hamiltonian for the programmable network with four different “programs” and one data qubit. If first two qubits have the state $|0\rangle|0\rangle$, the data qubit is not changed, but if they have state $|0\rangle|1\rangle$, $|1\rangle|0\rangle$, or $|1\rangle|1\rangle$ during a period τ , then the data qubit is changed as $|D\rangle \mapsto \exp(i\sigma_k\tau)|D\rangle$ with $k = 1, 2, 3$ respectively.

From the one hand, the example shows a specific kind of the exactly universal programmable quantum gate satisfying Eq. (12) with the finite-dimensional control register. Here the continuous parameter τ is used to implement the infinite number of different programs. From the other hand, for a programmable quantum network it is more appropriate to fix some period $\Delta\tau$ and use consequent application of gates $\mathbf{u}_k = \exp(i\sigma_k\Delta\tau)$ via the three-buses design discussed in Sec. 2.6.

Here is not suggested, that $\Delta\tau$ should be infinitesimally small. Using the infinitesimal parameter is not the only way of approach to a continuous limit. Most Hamiltonians used here generate gates with a simple periodic behavior due to Eq. (34) and if to choose $\Delta\tau$ as an *irrational* multiple of π , it is possible to approximate any real parameter with an arbitrary precision. Such an idea really was used already in the earliest works about the theory of universal quantum computations [1, 4, 15].

Using qubits for the control is convenient, but is not necessary. *E.g.*, it is enough for universality to use only two Pauli matrices, and so instead of two qubits it is possible to use one *qutrit* for ‘3→2’ control, *i.e.*, a quantum system with three states $|0\rangle, |1\rangle$ and $|2\rangle$ and to write a Hamiltonian like

$$\mathbf{h}_{\triangleright}^{\epsilon} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \begin{matrix} |0\rangle|0\rangle \\ |0\rangle|1\rangle \\ |1\rangle|0\rangle \\ |1\rangle|1\rangle \\ |2\rangle|0\rangle \\ |2\rangle|1\rangle \end{matrix}. \quad (45)$$

The very rough scheme of such a system is depicted on Fig. 10. Here a qubit — is a system with spin-1/2, and a qutrit is represented as a quantum system distributed in a potential of a triple quantum dot “molecule.” It is suggested, that in the state $|0\rangle$ the qutrit is not interact with the qubit, but the states $|1\rangle$ and $|2\rangle$ already affect on the spin system and produce some evolution of the qubit like $|D\rangle \mapsto \exp(i\boldsymbol{\sigma}_k\tau)|D\rangle$, $k = 1, 2$.

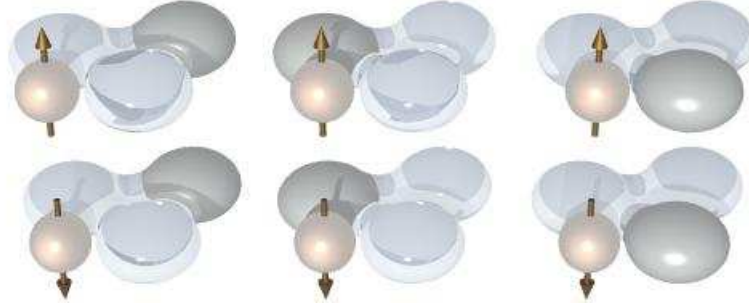


Figure 10: A scheme of six basic states for ‘3→2’ control Eq. (45). The qubit is depicted as a system with spin and the qutrit as a “triple quantum dot.”

Let us return to the controlled two-qubit gates Eq. (41). It is convenient for further applications to place the control qubit between the two controlled ones and then, rewriting Eq. (41), it is possible to consider a diagonal Hamiltonian with three energy levels

$$\begin{aligned} E &= 0 & : & |0\rangle|0\rangle|0\rangle, |0\rangle|0\rangle|1\rangle, |1\rangle|0\rangle|0\rangle, |1\rangle|0\rangle|1\rangle \\ E &= -\Delta E & : & |1\rangle|1\rangle|0\rangle, |0\rangle|1\rangle|1\rangle \\ E &= +\Delta E & : & |0\rangle|1\rangle|0\rangle, |1\rangle|1\rangle|1\rangle \end{aligned}. \quad (46)$$

The scheme has visual interpretation: if the control qubit is in the state $|0\rangle$, the two data qubits are not interact, but if the control qubit has the state $|1\rangle$, then the energy

of the system is bigger for the data qubits in same states and smaller for different ones. So the program qubit may be considered as a “control switch” of the interaction between the two data qubits. An illustrative, but a rather naïve scheme with a double quantum dot for the control of two spin systems is depicted on Fig. 11.

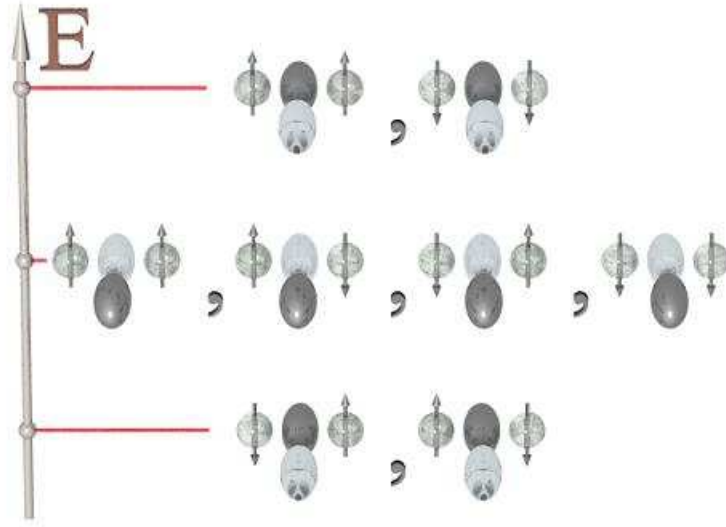


Figure 11: A scheme of energy levels for the diagonal Hamiltonian of control Eq. (46)

Some discussion on physical systems appropriate for such a “switching” purposes may be found also in papers related with models of the “global” quantum computing (with “always-on interactions”) [37, 38, 39, 40], because such approach is very close to the idea of programmable quantum networks.

Finally, a programmable network with such a kind of gates for one- and two-qubit operations is depicted on Fig. 12. Here a data register is presented as an array of spin-half systems and a control register is consisting of double and triple quantum dots. Double quantum dots used for two-qubits operation here are intermittent with data qubits, and qutrits are situated above.

The controlled one-qubit quantum gate for such a register may be described by Hamiltonians Eq. (40) and Eq. (45)

$$\mathbf{h}_{\triangleright}^c = |0\rangle\langle 0| \otimes \mathbf{1} + |1\rangle\langle 1| \otimes \boldsymbol{\sigma}_x + |2\rangle\langle 2| \otimes \boldsymbol{\sigma}_z. \quad (45')$$

It is enough simply to include number k of controlled qubit for such Hamiltonian $\mathbf{h}_{\triangleright}^{c(k)}$. On the considered scheme Fig. 12 for data qubits the number is $k = 2j + 1$.



Figure 12: A visual scheme of a programmable quantum register

Hamiltonians of controlled two-gates devote a bit more detailed consideration, because one data qubit is controlled by two different gates. Let us rewrite Eq. (42) for the control qubits between the two data qubits

$$\mathbf{h}_{\emptyset}^{\mathbf{c}} = \sigma_z \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes \sigma_z = \frac{1}{2} (\sigma_z^{(1)} \sigma_z^{(3)} - \sigma_z^{(1)} \sigma_z^{(2)} \sigma_z^{(3)}). \quad (47)$$

The analogue of decomposition Eq. (43) is

$$\begin{aligned} \mathbf{U}_{\emptyset}^{\mathbf{c}}(\tau) &= \exp(i\mathbf{h}_{\emptyset}^{\mathbf{c}}\tau) = \exp(i\frac{\tau}{2}\sigma_z^{(1)}\sigma_z^{(3)}) \exp(-i\frac{\tau}{2}\sigma_z^{(1)}\sigma_z^{(2)}\sigma_z^{(3)}) \\ &= \exp(i\frac{\tau}{2}\sigma_z^{(1)}\sigma_z^{(3)}) \exp(i\frac{\pi}{4}\sigma_z^{(1)}\sigma_x^{(2)}) \exp(-i\frac{\tau}{2}\sigma_y^{(2)}\sigma_z^{(3)}) \exp(-i\frac{\pi}{4}\sigma_z^{(1)}\sigma_x^{(2)}). \end{aligned} \quad (48)$$

For an array with qubits Hamiltonians Eq. (47) may be written as

$$\mathbf{h}_{\emptyset}^{\mathbf{c}(2k+1)} = \frac{1}{2} (\sigma_z^{(2k)} \sigma_z^{(2k+2)} - \sigma_z^{(2k)} \sigma_z^{(2k+1)} \sigma_z^{(2k+2)}). \quad (49)$$

The Hamiltonians Eq. (49) commute for different k and so despite of the overlap, again there is no an essential difference between the control of array Fig. 12 and the initial model with only three systems.

3.2.2 Shift gates

The models above represent only **C** (control) part of the **Control-Shift** network discussed earlier in Sec. 2.6. It is necessary to supply new and new indexes $|k\rangle$ for the control register and if control and data buses may be considered as some arrays along a given axis (x) Fig. 12, it is possible to arrange a program bus as some bar along an orthogonal axis (y) with a separate line (1D array) for each control gate and to produce the two-dimensional array Fig. 13.

In such 2D structure, each control element is receiving indexes from only one line, *i.e.*, 1D array along orthogonal axis (y , program). Let us consider necessary operations with the array.

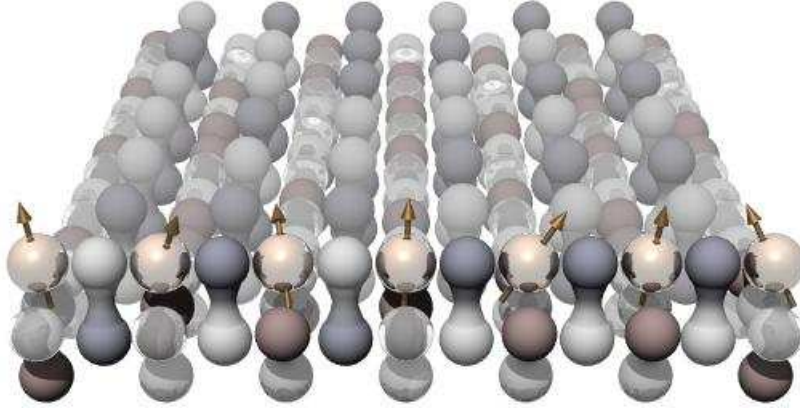


Figure 13: A two-dimensional programmable quantum network

The simplest idea — is to implement the cyclic shift used Eq. (25) for each such y -array

$$S: |k_L\rangle|k_{L-1}\rangle \cdots |k_2\rangle|k_1\rangle \mapsto |k_1\rangle|k_L\rangle|k_{L-1}\rangle \cdots |k_2\rangle. \quad (25')$$

Such a method produces valid realization of a programmable **Control-Shift** network, but there is a difficulty with implementation of the Hamiltonian for Eq. (25'), because it is *not a local* operation, *i.e.*, it acts on all L systems.

On the other hand all Hamiltonians for the control above were local with three systems or less. It is also possible to write local analogues of the **Shift** operation. The simple way — is to consider two different operations: for one step are exchanged all pairs $|k_{2j+1}\rangle$ and $|k_{2j+2}\rangle$ and for next step — pairs $|k_{2j+2}\rangle$ and $|k_{2j+3}\rangle$.

$$\begin{aligned} S_1: \cdots |k_{2j+1}\rangle|k_{2j+2}\rangle \cdots &\mapsto \cdots |k_{2j+2}\rangle|k_{2j+1}\rangle \cdots \\ S_2: \cdots |k_{2j+2}\rangle|k_{2j+3}\rangle \cdots &\mapsto \cdots |k_{2j+3}\rangle|k_{2j+2}\rangle \cdots \end{aligned} \quad (50)$$

So each step is performed using only the exchange (**SWAP**) operation with two systems. For qubits, qutrits, and quantum systems with an arbitrary number of states (“qudits”) the **SWAP** gate is defined on basis states as

$$\text{SWAP}: |k\rangle|l\rangle \mapsto |l\rangle|k\rangle. \quad (51)$$

For qubits it may be also implemented using three **c-NOT** gates [25]

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \text{c-NOT}_{12} \text{c-NOT}_{21} \text{c-NOT}_{12}. \quad (52)$$

If to use the alternating sequence of such operations, states with odd numbers like $|k_1\rangle$ are permanently shifted in one direction and states with even numbers — in opposite one. So it is possible to encode a necessary sequence if to use only the odd states and to set other to zero

$$|k_1\rangle|0\rangle|k_2\rangle|0\rangle\cdots \quad (53)$$

Here is suggested as usual, that the zero index corresponds to the identity operator.

3.3 Quantum cellular automata

The present paper is not devoted exclusively to the theory and applications of quantum cellular automata (QCA), but it is reasonable to discuss briefly the topic, because QCA models are widely used in many works devoted to related problems. It was already briefly mentioned *the global quantum computing* [37, 38, 39, 40]. Recent works on reversible quantum cellular automata [41, 42, 43] make possible to talk about even more direct relation between the model of universal programmable quantum networks with pure states and quantum cellular automata.

The just considered model Fig. 13 illustrates, that the **Control-Shift** design has very close relation with cellular automata, but there is specific subtleties for transition to the quantum case [41, 42].

Let us consider for example a spin lattice. It could be used formally for representation of a cellular automaton with two values $\{0, 1\}$ encoded by two basic states $|\uparrow\rangle$, $|\downarrow\rangle$ Fig. 14.



Figure 14: A lattice with qubits representing a cellular automaton

On the other hand, a general state of such a lattice is not necessary may be represented as a *product state* Fig. 15, then it may be simply treated as a collection of cells in different states. The general entangled state is defined as a sum Fig. 16 with complex coefficients on different basis configurations Fig. 14.

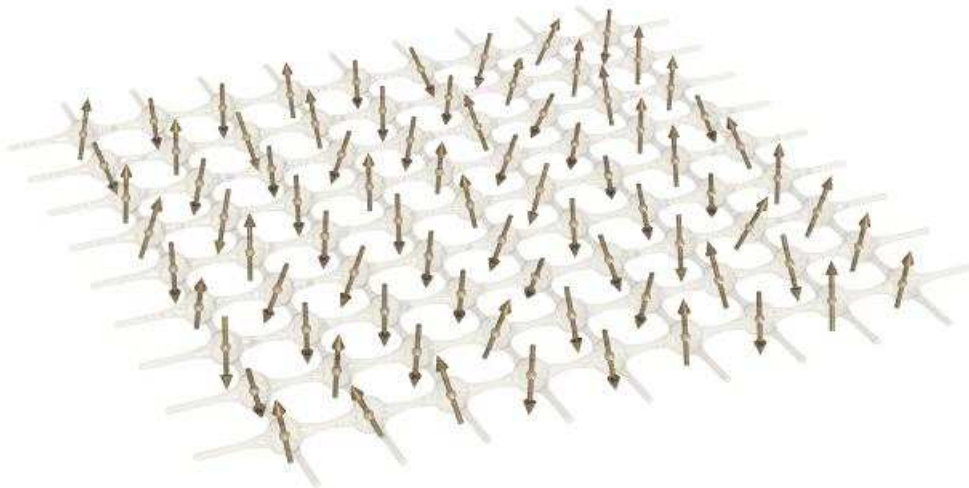


Figure 15: A lattice with qubits described by the product state

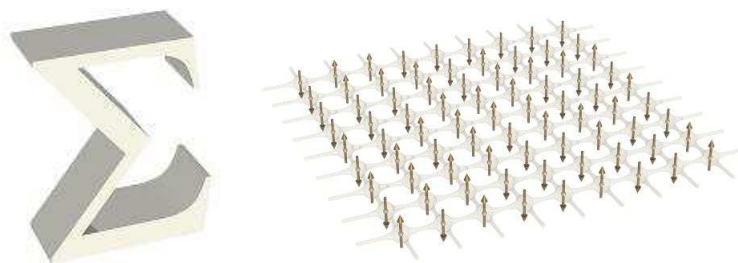


Figure 16: A sum on different configurations representing general entangled state

This specific property of a quantum system does not permit to talk about a state of a single cell, or a block of cells as about a vector in the Hilbert space, and it produces specific difficulties for local description of cellular automata.

On the other hand, it was already mentioned, that the program bus may be formally described using *reversible* classical computations and the theory of reversible cellular automata has a long history and is appropriate for such a purpose [44, 45, 46].

One specific method used in the theory of reversible cellular automata [45] is the *Margolus partition* with two different operations during different time steps. The simplest example for 1D automata just corresponds to Eq. (50) used for local implementation of the shift operation.

The alternating sequence of **S** and **C** operations in Eq. (27) used above in programmable **Control-Shift** networks in Sec. 2.6 also becomes a quantum analogue of the Margolus partition, if to consider 2D design of quantum gates described below Fig. 13 as a quantum cellular automaton.

The design considered here maybe still does not look like “true” QCA, because it is not regular enough, *e.g.*, there are 2D array for program with two different kinds of systems and 1D array of data. On the other hand, cellular automaton with different kinds of cells is equivalent to cellular automaton with only one kind with additional flag.

It is also useful to consider 2D array of data [43]. We just need to spread the whole programmable register Fig. 12 along program bus Fig. 17 (instead of spreading only control elements of the register depicted earlier on Fig. 13).

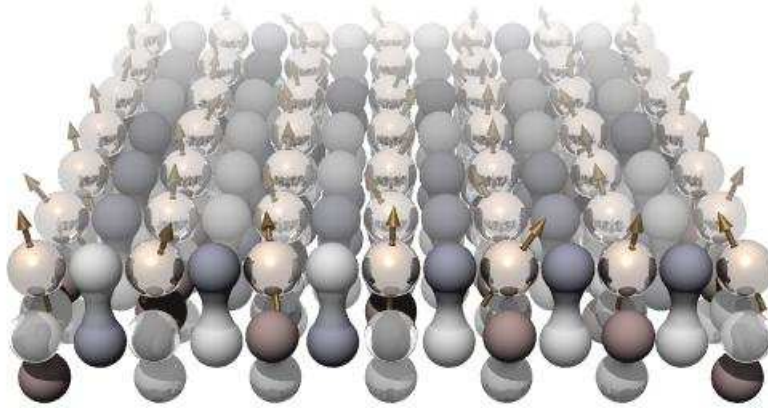


Figure 17: A quantum network with two-dimensional data and program registers

Such array may be considered as many independent copies of the programmable register. To produce initial design Fig. 13 from such a regular QCA, it is possible formally to introduce an additional flag-qubit and to modify the control register in such a way, that it changes state of data only if the flag is $|1\rangle$.

On the other hand, it is possible to save the initial structure and to let all program registers evolve in parallel. Implementation of the cyclic shift may be represented by cylindrical QCA Fig. 18.

Here we have three basic elements: the qubits of quantum data register(s), triple quantum dots for implementation of one-qubit gates (below a qubit on Fig. 12, Fig. 17,

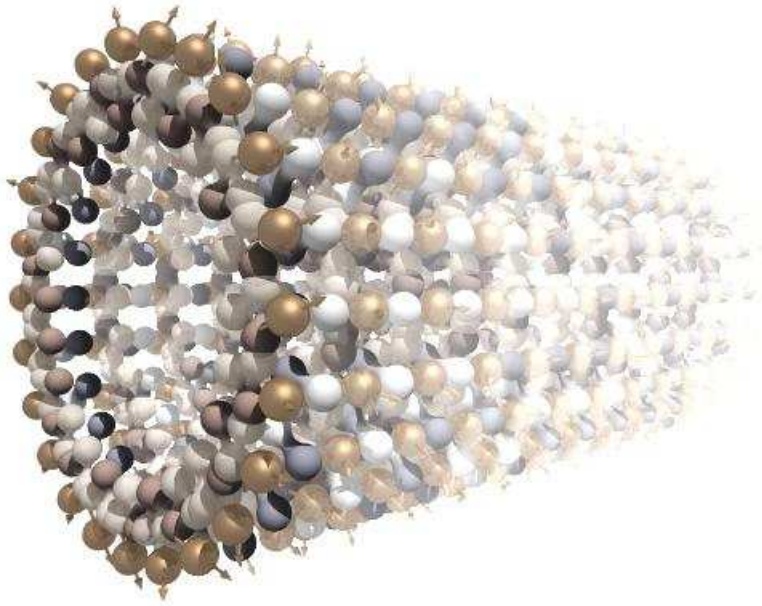


Figure 18: A programmable quantum network as cylindrical QCA

etc.), and double quantum dots for control of two-qubit gates (between each two qubits on Fig. 12, it should be mentioned, that on Fig. 17 the registers are arranged *along the axis* of the cylinder).

The QCA programmable network uses two standard steps **C** and **S** of Control-Shift network, but they are subdivided on smaller steps. The analysis of operation **S** is simpler due to direct relation with reversible classical computations already mentioned above.

The **S(hift)** steps move data around cylinder and may be implemented using two sub-steps S_1, S_2 Eq. (50). It was already mentioned, that the indexes in such a case should be intermittent with zeros Eq. (53).

Each control line may be considered independently and so **S** step is described by the expression Eq. (50). For implementation of the universal set of gates like Eq. (36) two different kinds of lines are necessary: for the control of two-qubit operations are used only two indexes $\{0, 1\}$ and for controlled one-qubit gates (at least for two first qubits) are used three indexes $\{0, 1, 2\}$.

The lines with indexes for one-qubit gates are alternating with lines for two-qubit gates. It is important to mention, that all two-qubit gates from Eq. (36d) are commuting and so may be applied on the same step. The one-qubit gates on different lines are also commuting. On the other hand, gates Eq. (36c) are not commuting with gates Eq. (36d) for adjoint lines, and so it is necessary to apply such gates on different steps.

It requires a special arrangement of nonzero indexes on adjoint lines.

Let us suggest that such arrangement condition is satisfied to prevent consideration of non-commuting Hamiltonians and consider **C(ontrol)** step.

One-qubit operations may be defined locally for each pair with the control qutrit and the data qubit. The Hamiltonian of control on such a pair was already discussed above Eq. (45') on the page 22. The Hamiltonians of controlled two-qubit gates again coincide with expressions for single controlled register Fig. 12 and also were considered earlier Eq. (49).

It was already mentioned, that the Hamiltonians Eq. (49) for control commute for different sites. The commutativity is important for quantum cellular automata models [41, 42]. Let us suggest, that an entire Hamiltonian of QCA may be expressed as a sum of all local Hamiltonians corresponding to local cells. Only for commutative operators such a method produces appropriate relation between global and local transition functions

$$\exp(i \sum_{k,j} \mathbf{h}^{k,j} \Delta\tau) = \prod_{k,j} \exp(i \mathbf{h}^{k,j} \Delta\tau), \quad \mathbf{U}_{\text{global}} = \prod_{k,j} \mathbf{U}_{\text{local}}^{k,j} \quad (54)$$

The Margolus partition used in the theory of reversible QCA [41] just uses two sets of commuting Hamiltonians. In the model considered here may be found even four such sets. The Margolus partition used for **S(hift)** operation is corresponding to the two operators $\mathbf{S}_1, \mathbf{S}_2$ Eq. (50) and already was discussed above, but **C(ontrol)** operation also contains two sets of Hamiltonians corresponding to one- and two-gates

$$\mathbf{H}_{\triangleright}^{\mathbf{c}} = \sum_{k,j} \mathbf{h}_{\triangleright}^{\mathbf{c}(2k-1,j)}, \quad \mathbf{H}_{\emptyset}^{\mathbf{c}} = \sum_{k,j} \mathbf{h}_{\emptyset}^{\mathbf{c}(2k,j)}. \quad (55)$$

Let us introduce two operators

$$\mathbf{C}_1 = \exp(i \mathbf{H}_{\triangleright}^{\mathbf{c}} \Delta\tau), \quad \mathbf{C}_2 = \exp(i \mathbf{H}_{\emptyset}^{\mathbf{c}} \Delta\tau), \quad (56)$$

with Hamiltonians of one- and two-gates respectively and $\Delta\tau$ is the fixed irrational multiple of π , then the programmable quantum network based on such QCA may be described by the periodic sequence of operators like

$$\mathbf{U}_{\text{VI}} = \mathbf{C}_1 \mathbf{C}_2 \mathbf{S}_1 \mathbf{C}_1 \mathbf{C}_2 \mathbf{S}_2, \quad \mathbf{U}_{\text{IV}} = \mathbf{C}_1 \mathbf{S}_1 \mathbf{C}_2 \mathbf{S}_2, \quad (57)$$

there the second operator \mathbf{U}_{IV} with only four terms is using alternating ('checker-board') arrangement, when after each step half of indexes are zeros and so only one operator between $\mathbf{C}_1, \mathbf{C}_2$ produces nontrivial result. It should be mentioned, that both operators Eq. (57) correspond to two steps of a program and so for the cyclical design with the perimeter $2L$, it is necessary repeat such operators L times.

4 Conclusion

In this paper was considered the theory of programmable quantum networks. They may be considered as quite appropriate models of future quantum processors. Such devices would not require the macroscopic equipment traditional for modern experimental research in the area of quantum information processing, because it is suggested to use quantum systems for the program, control and data.

Such architecture is also interesting from the pure theoretical point of view. It let us use the unified description of all processes necessary for functionality of such quantum devices and reduces amount of problems related with consideration of transition from classical to quantum domain.

It should be mentioned, that some methods used in this paper also may be found in the more general theory about relations between quantum and classical pictures [27, 47] and so application for the quantum information science is quite justified.

In the present paper were used programmable quantum networks with pure states (sometime also called ‘deterministic’) and it produces additional clarification and simplification. It is also actual because ‘probabilistic’ programmable quantum gate arrays already are presented quite completely in other works.

The theory of programmable quantum networks is still in the state of development and some new interesting branches may appear. For example universal programmable quantum computers by definition may perform any transformations with data, but it corresponds to arbitrary manipulations with a quantum system. Due to such a principle emphasized already in earliest works [4] there is no big difference between an universal programmable quantum network and a quantum robot [48] with wide range of possible operations.

Say, it is quite reasonable to consider the famous Wigner’s question about possibility of “self-reproducing units” using the theory of universal programmable networks. In fact, some preliminary analysis was already performed using both probabilistic [50] and deterministic [51] approach, but the intriguing problem most likely devotes further research.

Acknowledgments

Author is grateful to V. Bužek, R. Raussendorf and A. K. Pati for the interesting exchange.

References

- [1] D. Deutsch, “Quantum computational networks,” *Proc. R. Soc. London Ser. A* **425**, 73–90 (1989).
- [2] D. Aharonov, A. Y. Kitaev, and N. Nisan, “Quantum circuits with mixed states,” in *Proc. 30th of the Annual ACM Symp. on the Theory of Computing (El Paso, Texas, 1998)* 20–30 (ACM Press, New York, 1998); *Preprint quant-ph/9806029*.
- [3] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” in *Proc. of the 35th Annual Symp. on the Foundations of Computer Science (Santa Fe, New Mexico, 1994)* 124–134 (IEEE Computer Science Society Press, Los Alamitos, California, 1994).
- [4] D. Deutsch, “Quantum theory, the Church-Turing principle and the universal quantum computer,” *Proc. R. Soc. London Ser. A* **400**, 97–117 (1985).
- [5] A. Burks, H. Goldstine, J. v. Neumann, *Preliminary discussion of the logical design of an electronic computing instrument*, **4** (Princeton 1946).
- [6] M. A. Nielsen and I. L. Chuang, “Programmable quantum gate arrays,” *Phys. Rev. Lett.* **79** 321–324, (1997); *Preprint quant-ph/9703032*.
- [7] W. K. Wootters, W. H. Zurek, “A single quantum cannot be cloned,” *Nature* **299**, 802–803 (1982).
- [8] A. Yu. Vlasov, “Quantum Processors and Controllers,” in *Proc. Int. Conf. Physics and Control, PhysCon 2003*, (St-Petersburg, 2003), pp 861–866; *Preprint quant-ph/0301147*.
- [9] G. Vidal and J. I. Cirac, “Storage of quantum dynamics on quantum states: A quasi-perfect programmable quantum gate,” *Preprint quant-ph/0012067* (2000).
- [10] G. Vidal, L. Masanes, and J. I. Cirac, “Storing quantum dynamics in quantum states: stochastic programmable gate for $U(1)$ operations,” *Phys. Rev. A* **88**, 047905 (2002); *Preprint quant-ph/0102037*.
- [11] M. Hillery, V. Bužek, and M. Ziman, “Probabilistic implementation of universal quantum processors,” *Phys. Rev. A* **65**, 022301 (2002); *Preprint quant-ph/0106088*.
- [12] M. Hillery, M. Ziman, and V. Bužek, “Implementation of quantum maps by programmable quantum processors,” *Phys. Rev. A* **66**, 042302 (2002); *Preprint quant-ph/0205050*.

- [13] M. Hillery, M. Ziman, and V. Bužek, “Improving performance of probabilistic programmable quantum processors,” *Phys. Rev. A* **69**, 042311 (2004); *Preprint quant-ph/0311170*.
- [14] M. Ziman and V. Bužek, “Realization of POVMs using measurement-assisted programmable quantum processors,” *Preprint quant-ph/0411135* (2004).
- [15] D. Deutsch, A. Barenco, and A. Ekert, “Universality in quantum computation,” *Proc. R. Soc. London Ser. A* **449**, 669–677 (1995).
- [16] A. Yu. Vlasov, “Classical programmability is enough for quantum circuits universality in approximate sense,” *Preprint quant-ph/0103119* (2001).
- [17] A. Yu. Vlasov, “Universal quantum processors with arbitrary radix $n \geq 2$,” *Int. Conf. Quantum Information, ICQI 2001*; *Preprint quant-ph/0103127* (2001).
- [18] A. Yu. Vlasov, “Universal hybrid quantum processors,” *Part. Nucl. Lett.* **1**[116] 60–65 (2003); *Preprint quant-ph/0205074*.
- [19] A. Yu. Vlasov, “Von Neumann quantum processors,” in *304th WEH-Seminar ‘Elementary Quantum-Processors,’ Preprint quant-ph/0311196* (2003).
- [20] R. Cleve, “An introduction to quantum complexity theory,” in *Quantum computation and quantum information theory: Collected papers and notes (Villa Gualino, Torino, Italy, 1999)* ed. by C. Macchiavello, G. M. Palma, and A. Zeilinger (World Scientific, Singapore, 2000), pp 103–128; *Preprint quant-ph/9906111*.
- [21] S. Lloyd and S. L. Braunstein, “Quantum computation over continuous variables,” *Phys. Rev. A* **82**, 1784–1787 (1999); *Preprint quant-ph/981082*.
- [22] A. Barenco, D. Deutsch, A. K. Ekert, and R. Jozsa, “Conditional quantum dynamics and logic gates,” *Phys. Rev. Lett.* **74**, 4083–4086 (1995).
- [23] A. Yu. Vlasov, “Error correction with Euclidean qubits,” *Quantum Computers & Computing* **2**, 68–78 (2001); *Preprint quant-ph/9911074*.
- [24] C. M. Caves, C. A. Fuchs, and P. Rungta, “Entanglement of formation of an arbitrary state of two rebits,” *Found. Phys. Lett.* **14**, 199–212 (2001); *Preprint quant-ph/0009063*.
- [25] R. Feynman, “Quantum-mechanical computers,” *Opt. News* **11**, 2, 11-20 (1985); reprinted in *Found. Phys.* **16** 507–531 (1986).

- [26] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. W. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, “Elementary gates for quantum computation”, *Phys. Rev. A* **52**, 3457–3467 (1995); *Preprint quant-ph/9503016*.
- [27] W. H. Zurek, “Decoherence, einselection, and the existential interpretation (The rough guide),” *Philos. Trans. R. Soc. Lond. A* **356**, 1793–1821 (1998); *Preprint quant-ph/9805065*.
- [28] D. Beckman, D. Gottesman, M. A. Nielsen, and J. Preskill, “Causal and localizable quantum operations,” *Phys. Rev. A* **64**, 052309 (2001); *Preprint quant-ph/0102043*.
- [29] S. Lloyd, “Hybrid quantum computing,” *Preprint quant-ph/0008057*, (2000).
- [30] A. Peres, and D. R. Terno, “Hybrid classical–quantum dynamics,” *Phys. Rev. A* **63**, 022101 (2001); *Preprint quant-ph/0008068* (2000).
- [31] D. P. DiVincenzo, “Two-bit gates are universal for quantum computation,” *Phys. Rev. A* **51**, 1015 – 1022 (1995).
- [32] S. Lloyd, “Universal quantum simulators,” *Science* **273**, 1073 (1996).
- [33] A. Yu. Vlasov, “Clifford algebras and universal sets of quantum gates,” *Phys. Rev. A* **63**, 054302 (2001); *Preprint quant-ph/0010071*.
- [34] R. R. Tucci, “A rudimentary quantum compiler” (Second Ed.), *Preprint quant-ph/9902062* (1999).
- [35] V. V. Shende, I. L. Markov, and S. S. Bullock, “Minimal universal two-qubit CONTROLLED–NOT–based circuits,” *Phys. Rev. A* **69**, 062321 (2004); *Preprint quant-ph/0308033*.
- [36] M. Mottonen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, “Quantum circuits for general multiqubit gates,” *Phys. Rev. Lett.* **93**, 130502 (2004); *Preprint quant-ph/0404089*.
- [37] S. C. Benjamin, “Schemes for parallel quantum computation without local control of qubits,” *Phys. Rev. A* **61**, 020301 (2000); *Preprint quant-ph/9909007*.
- [38] S. C. Benjamin and S. Bose, “Quantum computing with an always-on Heisenberg interaction,” *Phys. Rev. Lett.* **90**, 247901 (2003); *Preprint quant-ph/0210157*.
- [39] S. C. Benjamin, “Multi-qubit gates in arrays coupled by ‘always on’ interactions,” *Preprint quant-ph/0403077* (2004).

- [40] S. C. Benjamin, B. W. Lovett, and J. H. Reina, “Optical quantum computation with perpetually coupled spins,” *Phys. Rev. A* **70**, 060305 (2004); *Preprint quant-ph/0407063*.
- [41] B. W. Schumacher and R. F. Werner, “Reversible quantum cellular automata,” *Preprint quant-ph/0405174* (2004).
- [42] A. Yu. Vlasov, “On quantum cellular automata,” *Preprint quant-ph/0406119* (2004).
- [43] R. Raussendorf, “A quantum cellular automaton for universal quantum computation,” *Preprint quant-ph/0412048* (2004).
- [44] T. Toffoli, “Computation and construction universality of reversible cellular automata,” *J. Comp. Syst. Sci.* **5**, 213–231 (1977).
- [45] T. Toffoli and N. Margolus, “Invertible cellular automata: a review,” *Physica D* **45**, 229–253 (1990).
- [46] S. Wolfram, *Cellular automata and complexity : Collected papers*, (Addison-Wesley, Reading MA 1994).
- [47] W. H. Zurek, “Decoherence, einselection, and the quantum origins of the classical,” *Rev. Mod. Phys.* **75**, 715–775 (2003); *Preprint quant-ph/0105127*.
- [48] P. Benioff, “Quantum robots and environments,” *Phys. Rev. A* **58**, 893–904 (1998); *Preprint quant-ph/9802067*.
- [49] E. P. Wigner, “The probability of the existence of a self-reproducing unit,” in *The Logic of Personal Knowledge. Essays in Honor of Michael Polanyi*, (Routledge and Kegan Paul, London, 1961); reprinted in E. P. Wigner, *Symmetries and Reflections*, (Indiana University Press, Bloomington, Indiana, 1967).
- [50] A. K. Pati and S. L. Braunstein, “Quantum mechanical Universal constructor,” *Preprint quant-ph/0303124* (2003).
- [51] A. Yu. Vlasov, “Nondeterministic recursion with quantum units,” *Preprint quant-ph/0403037* (2004).